2014 47th Annual IEEE/ACM International Symposium on Microarchitecture

Hi-Rise: A High-Radix Switch for 3D Integration with Single-cycle Arbitration

Supreet Jeloka, Reetuparna Das, Ronald G. Dreslinski, Trevor Mudge, David Blaauw
University of Michigan, Ann Arbor
{*sjeloka,reetudas,rdreslin,tnm,blaauw*}*@umich.edu*

***Abstract*—This paper proposes a novel 3D switch, called '*Hi-Rise*', that employs high-radix switches to efficiently route data across multiple stacked layers of dies. The proposed interconnect is hierarchical and composed of two switches per silicon layer and a set of dedicated layer to layer channels. However, a hierarchical 3D switch can lead to unfair arbitration across different layers. To address this, the paper proposes a unique class-based arbitration scheme that is fully integrated into the switching fabric, and is easy to implement. It makes the 3D hierarchical switch's fairness comparable to that of a flat 2D switch with least recently granted arbitration.**

The 3D switch is evaluated for different radices, number of stacked layers, and different 3D integration technologies. A 64-radix, 128-bit width, 4-layer *Hi-Rise* evaluated in a 32nm technology has a throughput of 10.65 Tbps for uniform random traffic. Compared to a 2D design this corresponds to a 15% improvement in throughput, a 33% area reduction, a 20% latency reduction, and a 38% energy per transaction reduction.

***Keywords*-3D Integration; High-Radix Switch; Arbitration;**

I. Introduction

The number of cores on a single chip has seen a steady upward trend due to emerging parallel workloads and the need to meet performance goals within constrained power budgets. These many-core systems require low latency, area-energy efficient interconnects with extremely high bandwidth. Conventional interconnects constructed out of low-radix switches such as a *2D-Mesh* [1], [2], do not scale well because of the decreased performance resulting from larger hop counts and high power consumption [3]. Therefore, an interconnect fabric with efficiently designed high-radix switches is optimal for future many-core processors [4], [5].

Concurrently, 3D integration has become an important means of improving performance as process scaling slows down. This technique allows the number of cores to be increased by stacking different layers [6], with short vertical connections between the layers. These short connections can be leveraged for speeding up inter-layer communication and building an efficient interconnect. Interconnects based on low-radix 3D switches [7], [8], [9], [10] have been proposed in the past for 3D multi-core processors. However, as previously mentioned, low-radix and low bandwidth switches do not provide good scalability for a large number of cores.

3D high-radix switch design entails its own unique challenges. Unlike a flat 2D high-radix switch [11], the inputs and outputs of a 3D switch are spread over multiple layers. A 3D high-radix switch requires both intra-layer connections and inter-layer connections. The inter-layer vertical connections between silicon layers are made using Through-Silicon Vias (TSV). This leads to a heterogeneity in the intra-layer and inter-layer connections. Consequently, a simple 3D high-radix [12] switch folded over silicon layers has lower performance than a flat 2D switch. A high-radix 3D switch design thus requires: 1) switch datapath optimized for this connection heterogeneity; 2) composable and fair arbitration scheme across inter-layer and intra-layer connections; 3) reduction in the number of expensive TSVs with minimal impact on the switch performance parameters, i.e., throughput, latency and fairness; and, 4) improved area and energy efficiency to offset the design and manufacturing cost.

This paper proposes *Hi-Rise*, a high-radix 3D switch that achieves significant scalability and reduces the required number of TSVs by using a hierarchical architecture with dedicated layer-to-layer channels. The proposed *Hi-Rise* switch is divided into layers, each layer has two switches, a local switch and an inter-layer switch. The local switch connects local inputs to both intermediate outputs and vertical channels to other layers. The inter-layer switch connects both vertical channels from other layers and the intermediate outputs from the local switch to the final outputs on its layer. When combined, the two switches per layer result in a fully connected switch.

The hierarchical datapath of the switch is optimized for 3D connections. A key issue with the hierarchical switch datapath is that it can lead to unfairness as the arbitration is decomposed into two phases. To address this, we propose a new arbitration scheme, Class-based Least Recently Granted (CLRG), which brings the fairness of a hierarchical 3D switch close to that of a flat 2D switch using LRG priority. In this scheme, the inter-layer switch maintains a small counter for each input which signifies that input's output usage, and accordingly bins the requestors into different priority classes. Inputs in the same class use LRG to break ties. In contrast to CLRG, the implementation complexity of prior multi-stage arbiter designs [13], [14] make them unattractive for high-radix switches. In addition, these arbiter designs are not optimized for 3D, and lead to high inter-layer traffic, unlike the proposed CLRG scheme. We demonstrate that the proposed class based arbitration allows for single cycle arbitration and full integration within the switch fabric, with no area and negligible performance overheads.

1072-4451/14 $31.00 © 2014 IEEE
DOI 10.1109/MICRO.2014.45

The proposed 3D switch is evaluated for various architectural and physical configurations. We study the proposed switch design through detailed circuit-level delay analysis, power modeling, and micro-architectural cycle accurate performance simulations. We study various synthetic traffic patterns, and also real application benchmarks. The 3D switch is analyzed for different radices, number of stacked layers, and different TSV technologies. A 64-radix, 128-bit, 4-layer *Hi-Rise* is evaluated in detail using a 32nm technology. It has a throughput of 10.65 *Tbps* for uniform random traffic, which marks a 15% improvement over a 2D design along with a 33% area reduction, 20% latency reduction, and 38% energy per transaction reduction. For application workloads evaluated on a 64-core processor, *Hi-Rise* switch improves overall performance by 8% on average over a 2D switch.

In summary, our key contributions are:

- *Hi-Rise*, an efficient 3D high-radix switch. The proposed switch is a true 3D switch which connects inputs and outputs across different silicon layers.
- *Hi-Rise* adopts a hierarchical architecture with two internal switches per layer and dedicated layer-to-layer channels, to improve area efficiency, lower delay, and minimize the number of inter-layer TSVs.
- *Hi-Rise* provides built-in single-cycle arbitration across all inputs and outputs across different silicon layers. This improves efficiency and scalability.
- We propose a new class based arbitration scheme that is fully integrated into the switching fabric. This scheme makes the 3D hierarchical switch's fairness comparable to that of a flat 2D switch.
- At a radix of 64, *Hi-Rise* achieves an operating frequency of $2.2GHz$, consumes $44pJ$ of energy per 128-bit transaction and has an area of $0.451mm^2$ in 32 nm technology. The proposed switch extends scalability to radix 96 from that of the 64 radix supported by 2D switches at the same operating frequency.

II. Background

A. *The 2D* Swizzle-Switch

This section provides a brief background of a high-radix 2D *Swizzle-Switch* [11], [15]. As discussed earlier, unlike a 2D flat switch, a high-radix 3D switch design connects inputs and outputs across multiple layers with both *intra-layer* and expensive vertical TSV *inter-layer* connections. Our proposed *Hi-Rise* switch solves the design challenges of a 3D switch, while using the basic concepts of a 2D *Swizzle-Switch* for its internal switch structures.

A 2D *Swizzle-Switch* is a matrix type crossbar, with built-in arbitration, optimized for high radix switches. The input and outputs of the switch are placed in a grid fashion. The intersection of the horizontal input bus, with the vertical output bus is termed as a *cross-point*. A cross-point contains a connectivity bit, which if set, connects its input and output bus. The connectivity bit is set during the arbitration phase. The cross-point also stores a priority vector, containing priority information of its input with respect to all other inputs, for this output. The priority vector is updated based on LRG priority at the end of the arbitration phase.

The arbitration phase begins with each input requesting the outputs with which it wants to communicate. The input data lines are reused to index the outputs during arbitration. The output data lines are also reused as a priority bus during the arbitration phase. One advantage of reusing the output bus for priority lines during arbitration is that the same hardware used for data transfer (pre-charge, pull-down drivers and sense-amps) are reused during arbitration. This allows arbitration to be incorporated into the switch fabric without additional area overhead (since space underneath the cross-point is otherwise largely unused), and guarantees that the arbitration delay is identical to the datapath delay.

Thus, by embedding the logic-dominated arbitration into the wire-dominated crossbar, the 2D *Swizzle-Switch* allows a compact design and scaling of matrix crossbars to high radices.

B. *Baseline 3D Switch: A Folded 2D Switch*

A natural extension of the 2D switch to a 3D stacked implementation is to fold the 2D switch over multiple silicon layers. For this, the inputs and outputs will be redistributed across the layers.

A 64×64 2D switch evenly folded across four layers, will result in 16 inputs and 16 outputs on each layer. Since each input still needs to be able to communicate with all outputs, each layer will have a 16×64 switch, with 16 outputs connected locally as shown in Fig. 1. Note that there are still 64 output buses running from layer 1 through layer 4, and that each layer has a cross-point for *all* 64 outputs. Vertical TSVs go down from each layer to connect the 64 outputs lines between the layers. Essentially, the switch is a single 64×64 switch that is *folded* in the y-dimension across the layers.

In this basic 3D switch design each *layer* in itself has some benefit in terms of compactness, as instead of 64 nodes each layer has only 16 nodes. This compactness is an inherent benefit of 3D stacking. However, the delay of the switch itself is increased, as shown in Table I, because the wire and device capacitance in the switch remains the same after folding, while the addition of TSVs add to the total capacitance. Also, the number of TSVs required is very high as every output bus wire has to reach every layer.

Overall the folded switch configuration has higher implementation costs in terms of silicon area, delay, and energy over the 2D switch. The folded configuration was proposed and evaluated by Sewell et al. [12], however, their calculations incorrectly identified the number of TSVs required, and switch delay. Table I reflects the correct

Table I. IMPLEMENTATION COST OF 2D VERSUS 3D FOLDED SWITCH IMPLEMENTATIONS FOR 64-RADIX. THE 3D SWITCH HAS 4-LAYERS.

Design	Configuration	Area (mm^2)	Frequency (GHz)	Energy/transaction ($pJ/trans$)	Throughput ($Tbps$)	#TSVs
2D	64x64	0.672	**1.69**	71	9.24	0
3D Folded	[16x64]x4	0.705	**1.58**	73	8.86	8192

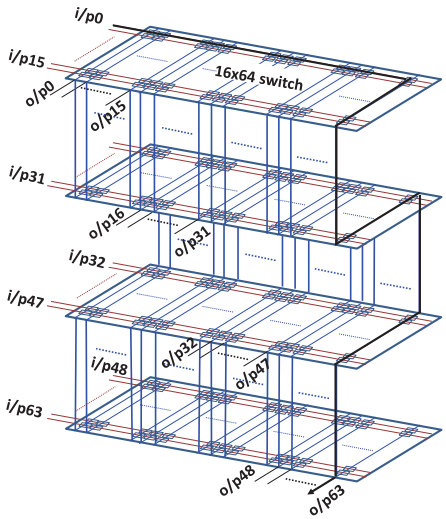

Figure 1. 3D Folded switch. A 64-radix 3D switch connecting four layers, each layer has 16-inputs and 16-outputs.

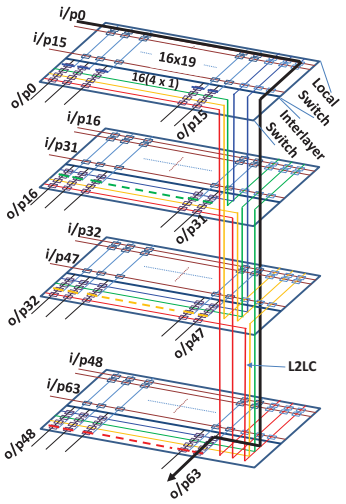

Figure 2. Conceptual view of *Hi-Rise* Switch. Blue L2LCs from L1 connects to L2, L3 and L4. Similarly, green connects L2, yellow connects L3 and red connects L4. The dark line shows i/p 0 connecting to o/p 63 through the local-switch on L1, down the L2LC, and then through the inter-layer switch on L4

calculations. The folded baseline is still a single, radix-64 switch, which requires 64x64 cross-points, unlike the much leaner proposed *Hi-Rise* switch with hierarchical datapath. The arbitration in a 3D folded switch is *identical* to that of a 2D *Swizzle-Switch*, whereas our proposed *Hi-Rise* switch has a *two phase class-based arbitration* which composes fairly over inter-layer and intra-layer connections. Our goal is to realize a 3D switch with significantly improved efficiency that takes advantage of the potential that 3D integration affords.

III. *Hi-Rise*: 3D SWITCH ARCHITECTURE

First we focus on the datapath of our proposed *Hi-Rise* switch followed by details of the arbitration mechanisms in Section III-B.

A. 3D Switch Datapath

For a switch with radix N, the *Hi-Rise* switch divides the N inputs and N outputs equally amongst the L layers of stacking. Therefore, at each layer we have N/L inputs and N/L outputs. It provides one or more dedicated vertical layer-to-layer channels (L2LC) from each layer to all other $L-1$ layers, as shown in Fig. 2.

To create a fully connected switch, each input on any layer must be able to arbitrate for, and transmit data to all outputs i.e. outputs on the same layer, and also outputs on every other layer. For this, each layer has two blocks, as shown in Fig. 3. The first block on a layer, referred to as the *local switch*, allows inputs to arbitrate for both local intermediate outputs on its layer, and outgoing vertical L2LCs to reach other layers. The second block, the *inter-layer* switch, is made up of several sub-blocks. Each inter-layer switch sub-block arbitrates between one particular incoming local intermediate output, and all the incoming vertical L2LCs from other layers, and forms the connection to one final output.

We define *channel multiplicity* as the number of L2LCs between any two layers, which we denote by the variable *'c'*. The local switch, has N/L inputs, and both intermediate outputs (N/L) and vertical L2LC outputs $(c\cdot(L-1))$. The local switch handles requests from all N/L inputs on a layer, and routes them to the desired layer, which may also be the current layer. The inter-layer switch on a layer has N/L sub-blocks. Each sub-block can connect a unique output to either one of the $(c\cdot(L-1))$ vertical channels coming in from other layers or to the unique intermediate output from the local switch on its own layer.

As an example, a 64-radix switch spread across 4 layers of silicon, will have 16 inputs and 16 outputs on each layer for the proposed 3D configuration. If $c=1$, i.e. there is only one L2LC between any two layers, as shown in Fig. 2, then the local switch is a 16×19 switch and the inter-layer switch

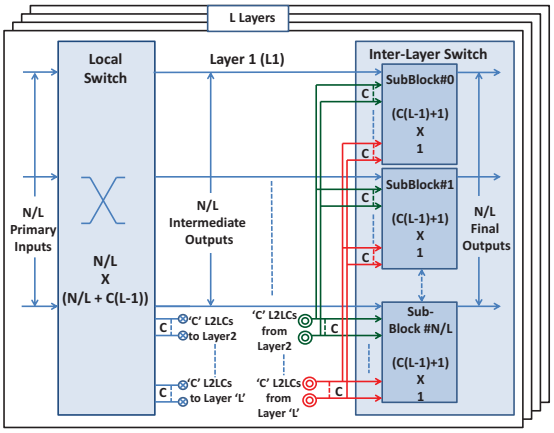

Figure 3. One Layer of a generic NxN L-layered *Hi-Rise* switch with Channel Multiplicity of 'c'. Each connecting line in the figure is a bus of width equal to flit size. Note- Each sub-block gets a unique output from the local switch, but the sub-blocks share the same bus from other layers.

has 16 sub-blocks of 4×1. In the inter-layer switch each final output can choose from 4 inputs—the three incoming L2LCs from the other three layers or the intermediate output bus from the local switch.

Suppose input 0 from layer 1 ($L1$) wants to communicate to output 63 on layer 4 ($L4$). Input 0 has to first win the dedicated L2LC from L1 to L4, by competing against all inputs from its own layer ($L1$) that want to communicate to $L4$. This arbitration happens on the local 16×19 switch on $L1$. Upon winning this L2LC, input 0 gets access to the inter-layer switch on $L4$. It has to then contend against similar winners from $L2$ and $L3$ wanting to communicate with output 63, and also the local contender from within $L4$ on intermediate output 63. Once the connection has been setup, input 0 can transmit flits across the layer to output 63, using the L2LC.

In the previous example configuration, there was only one vertical L2LC between any two layers. This L2LC is required to service any request from the inputs on L1 to the outputs on L4. In the absence of a strong spatial locality the vertical L2LCs can limit inter-layer traffic, and become bottlenecks. This problem can be solved by increasing the channel multiplicity 'c'. However, the addition of more L2LCs, leads to increased size of both the local switch and the inter-layer switch. The outputs on the local switch grow by $L-1$ for every additional channel. The number of inputs on the inter-layer switch also grows similarly by $L-1$. For the previous 64-radix example, a switch with $c = 4$ will have a 16×28 local switch, and 16 sub-blocks of 13×1 on the inter-layer switch. For channel multiplicity greater than one, rules are needed to allocate inputs to L2LCs. We discuss below a few possible channel allocation policies.

- **Input Binned:** The inputs on a layer are given a fixed, uniform allocation to the L2LCs. In this case, in N radix with L layers and a channel multiplicity of c, each L2LC will service request from $N/(L \times c)$ pre-assigned inputs. These inputs are selected in an interleaved fashion to reduce spatial locality dependence.
- **Output Binned:** Output binned is similar to input binning, except it is based on the output.
- **Priority Based:** The above two methods of channel allocation may lead to under utilization of the critical vertical L2LCs under certain adversarial traffic as the assignments are fixed. A more efficient utilization can be done by using a priority mux to choose between all N/L inputs However this method incurs higher delay because arbitration across L2LCs is now serialized.

B. 3D Switch Arbitration

In this section we discuss the arbitration mechanisms employed for our proposed *Hi-Rise* switch architecture. Both the local switch and the sub-blocks of the inter-layer switch can have different arbitration schemes that trade overall throughput and design complexity for fairness. The motivation of these schemes is to get as close as possible to the fairness of a 2D flat switch using a Least Recently Granted (LRG) scheme.

1) Baseline Layer-to-Layer (L-2-L) Priority: This approach applies a simple, independent LRG policy on both the switches on a layer. For a 64-radix, 4-layered switch, the local switch has 16 inputs. The local switch thus maintains a 16-bit LRG priority vector at each cross-point, to arbitrate only between the local inputs to win a local intermediate output channel or an L2LC. Each sub-block on the inter-layer switch will get as inputs, the L2LCs from each of the other three layers as well as an intermediate output on its layer; hence it only needs a small priority vector. For a 4 layer switch with channel multiplicity of one, a 4:1 LRG arbitration is required on each of the inter-layer sub-blocks.

The inter-layer switch follows the standard procedure; its priority is updated after every arbitration cycle. The priorities are updated at the local-switch only if it wins the final output. The local switch priority update is triggered by the winner at the inter-layer switch, and is back-propagated to the winner's local switch. This ensures that an input request always gets serviced, as its priority will rise on the inter-layer switch in subsequent arbitrations while remaining at the same priority on the local switch, thus, avoiding the possibility of starvation.

2) Unfairness with Baseline L-2-L LRG Priority: The baseline arbitration performs well for uniform random traffic. But, as the traffic requests to a particular output becomes more biased from a particular layer, the latency to service requests can become long.

We will illustrate this with the following example: a 1-channel 4-layer 64-radix configuration where 4 inputs, {3, 7, 11, 15} from the first layer (L1), and one input, {20} from the second layer (L2), are all requesting output 63 on layer four (L4). As shown in Fig. 4, all four inputs requesting

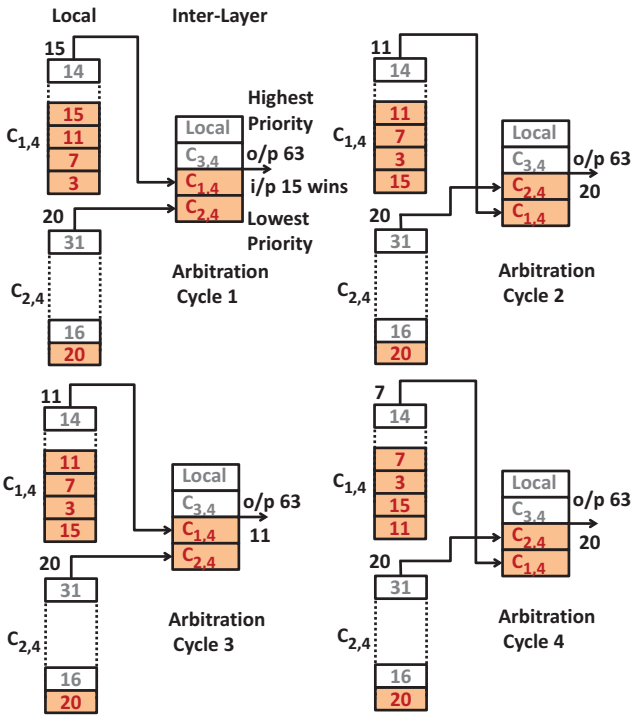

Figure 4. Baseline Layer-to-Layer (L-2-L LRG) Example. Only the inputs in red (shaded box) are requesting. The local switch winners contend at the inter-layer switch. The incoming L2LCs are designated as $C_{source\text{-}layer,destination\text{-}layer}$. L-2-L LRG allocates disproportionately to i/p 20

from L1 use the same L2LC going from L1 to L4, denoted as $C_{1,4}$. The four inputs compete with each other during arbitration at L1's local switch. On the other hand, input 20 is the only requester from L2, thus it always wins the local arbitration on its layer for $C_{2,4}$. At the sub-block of the inter-layer switch on L4 belonging to output 63, the winners of the local switch arbitrations compete for the final output.

Fig. 4 walks through four cycles of arbitration for this example. The LRG priorities decrease from top to bottom. In arbitration cycle 1, input 15 wins $C_{1,4}$ on the L1 local switch. Input 20, which is the lone contender for $C_{2,4}$, wins on the L2 local switch. The two local winners contend on the inter-layer switch of L4 where output 63 is being arbitrated. Input 15 wins as $C_{1,4}$ has higher priority than $C_{2,4}$. This is followed by an LRG update at both the sub-block of the inter-layer switch L4, and at the local switch on L1.

In the subsequent arbitration cycle, input 11 and input 20 contend but input 20 wins, as $C_{2,4}$ now has higher priority than $C_{1,4}$. The LRG of $C_{1,4}$ remains unchanged as it did not win at the inter-layer switch. Thus, in the next arbitration cycle input 11 again gets to contend and wins against input 20. The pattern continues, with one input amongst the four contenders on L1 winning, followed by the only contender from L2. The connections formed over time at output 63 is {15, 20, 11, 20, 7, 20, 3, 20, 15, 20 ...}. This pattern shows that the layer with the fewest number of contenders is able to access the output more frequently, and the arbitration is unfair. In a 2D flat switch with LRG the output pattern would be {20, 15, 11, 7, 3, 20, 15 ...}. The observation is that the baseline L-2-L LRG arbitration will be unfair, whenever multiple L2LCs contending for a single output have disparate number of requestors.

3) Weighted LRG Priority: To resolve the unfairness problem, the arbitration policy in the sub-blocks of the inter-layer switch needs to be modified. Weighted LRG (WLRG) arbitration scheme is a possible solution and is based on the intuition that L2LCs with higher traffic need to have higher priority. This can be achieved by freezing the LRG priorities for multiple cycles on the inter-layer switch sub-block when an L2LC has more than one requestor. The proportion of arbitration cycles for which the LRG is held, the weights, is determined by the number of requestors the L2LC represents.

Weights are generated by the local switch by counting the number of requestors. Weight information is then transmitted from the local switch to the inter-layer switch along with the request vector, and stored in a counter.

Calculating the number of requestors, involves counting the number of parallel requestors for an L2LC, which is hard to implement in hardware in a single-cycle. It makes the arbitration phase much longer, and hence slows down the cycle time for WLRG considerably. Furthermore, for a 3D switch the WLRG scheme becomes prohibitive due to the large amount of information (weights) that needs to be transmitted from the local switch to the interlayer switch over the L2LC.

4) Class-based Least Recently Granted (CLRG) Priority: To improve the fairness over the baseline L-2-L LRG without having to compute and transmit weights to the inter-layer switch, CLRG priority scheme is proposed.

In this scheme, at the inter-layer switch a counter is maintained for *each* input-output pair. By keeping track of all inputs (across all layers) at the inter-layer sub-block cross-points, the switch can be made fair. This counter tracks the number of times the specific primary input won the arbitration for a particular final output. The arbitration scheme at the inter-layer switch uses this count as a coarse priority, dividing the inputs into different subsets called *classes*. A bigger count value for an input signifies that the input has had a larger share of the bandwidth for this output, and it is relegated to a lower priority class. The inter-layer switch thus allows the contender with the least count to win. However, if input contenders belong to the same class, CLRG uses layer-to-layer LRG for tie-breaking.

To keep the counter based arbitration logic small, and to avoid cases where bursty traffic penalizes an input for a long time after the burst, the counter is kept short. The number of classes (counter length) required is a heuristic that needs to be tuned.

Whenever any counter saturates in a sub-block on the inter-layer switch, all 64 input counters for that sub-block are divided by 2. This maintains the relative class ordering between inputs.

Revisiting the 1-channel example, the counters of all the inputs will be initialized to 0, placing them in the highest priority class 'P0' as shown in Fig. 5. In arbitration cycle one, input 20 is the only contender from L2's local switch, and hence wins $C_{2,4}$ (the L2LC to layer L4). The LRG at L1's local switch has input 15 as the highest priority requesting input, and hence input 15 wins the arbitration for $C_{1,4}$. As both input 20 and input 15 are in priority class P0, LRG is used to tie-break. Input 20 wins, as $C_{2,4}$ has higher LRG priority than $C_{1,4}$. On winning the arbitration, input 20 increments its counter and moves to the lower priority class P1. In arbitration cycle 2 input 15 again contends against input 20. This time input 15 has class P0 and input 20 has class P1, therefore the switch employs class-based priority to make input 15 the winner. Even though LRG is not used for this arbitration cycle, it is still updated.

In arbitration cycle three, input 11 and input 20 contend and input 11 wins by virtue of its class, even though input 20 has a higher LRG priority. This is followed by input 7 and input 3 winning against input 20 as they are in class P0, while 20 is in P1. Now all requesting inputs have a count of 1, i.e., all are in class P1. In arbitration cycle 6, input 20 wins again on the basis of LRG tie-breaking. The sequence of winners for the class based arbitration will be {20, 15, 11, 7, 3, 20, 15, 11, 7, 3, 20 ...}.

This is similar to the pattern that will be followed in a single flat 2D LRG switch. Therefore, this scheme is able to resolve the fairness issue of the baseline scheme and also has an efficient single cycle hardware implementation, as we will see in the next section.

IV. Implementation

To design a high-radix switch with area-energy efficiency, the local and inter-layer sub-blocks are designed similar to a *Swizzle-Switch*. *Swizzle-Switch* is implemented by tiling together cross-points. This section details the cross-point design for both the local switch, and the inter-layer sub-blocks. It also details how we integrate the proposed CLRG logic within a cross-point.

A. Basic Cross-point Design

Recall, a cross-point connects an input to an output port and each cross-point contains both the connectivity and the arbitration logic. A 2D switch of radix N contains $N \times N$ cross-points, where each cross-point has a N-bit priority vector. The proposed *Hi-Rise* switch has *three* types of cross-points, intermediate output cross-points on the local switch, L2LC output cross-points on the local switch, and cross-points on the inter-layer sub-blocks. The underlying circuit for all three types of cross-points is similar to that of a 2D

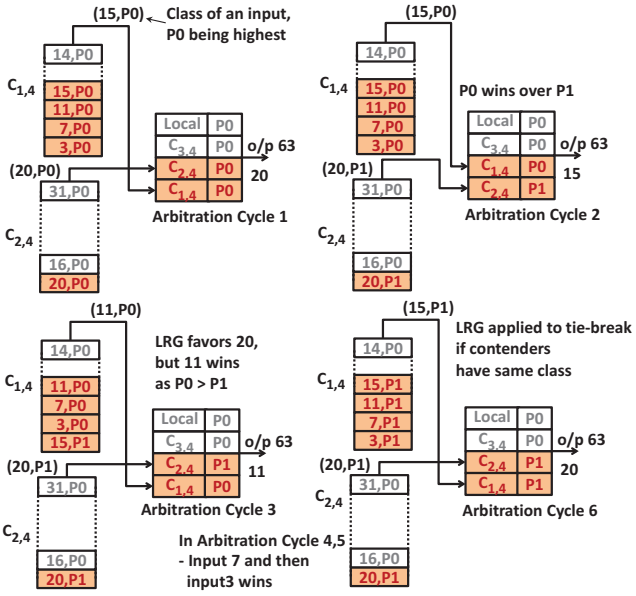

Figure 5. Class-based Least Recently Granted (CLRG) Arbitration Example. Class is based on the short-term history of input-output connections formed. Class information is maintained at the inter-layer switch.

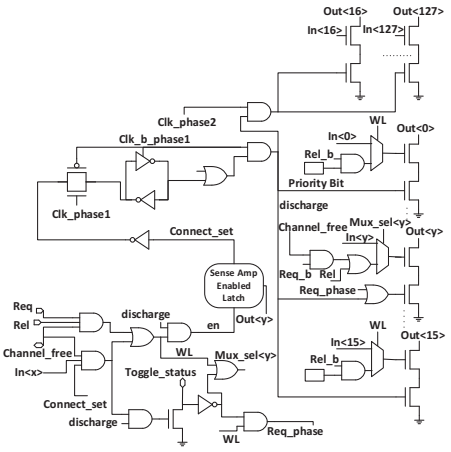

Figure 6. Circuit schematic of local intermediate output Cross-point. Out <15:0> reused for arbitration

switch. The circuit schematic for the first type of cross-point, the intermediate output cross-point, is shown in Fig. 6. The only difference from a 2D switch cross-point is that it has a N/L-bit priority vector, and two extra output bit-lines to transmit the request and the release signals to the inter-layer switch.

The second type of cross-point, L2LC output cross-point, builds upon the intermediate output cross-point. The L2LC cross-point differs from the intermediate output cross-point in two respects. First, the priority vector size is N/L in the intermediate output cross-point, whereas, it is only $(N/(L*c))$ for the input binned L2LC cross-point. Second, the L2LC cross-point transmits the request vector of the input to the

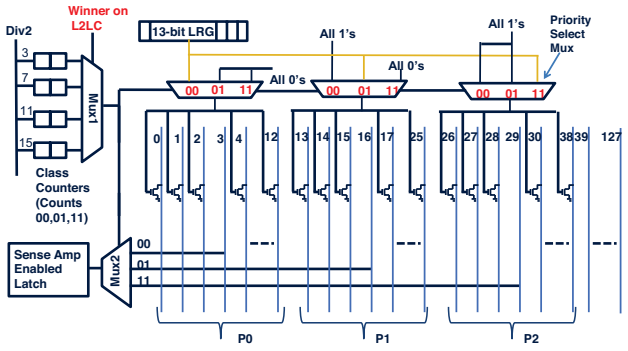

Figure 7. Conceptual view of a Cross-point at Inter-Layer Sub-block for CLRG arbitration. This cross-point is for an input binned 4-channel 4-layer 64-radix *Hi-Rise* switch

inter-layer switch during the arbitration phase, by setting high the L2LC wire with the requested output's index. This is because the L2LC output can request any of the N/L outputs on the destination inter-layer switch. On the other hand, the local intermediate output is dedicated to a single final output on the inter-layer switch.

The third type of cross-point, inter-layer sub-block cross-point, is discussed in the next section.

B. Arbitration Specific Cross-Point Design

The inter-layer sub-block cross-point structure is dependent on the arbitration scheme employed. For the baseline L-2-L LRG, the inter-layer cross-point is very similar to the basic 2D cross-point with only the priority vector size changed. Below we discuss the implementation of the Class-based LRG cross-point.

1) CLRG Cross-Point Design: The CLRG technique does not require any additional logic in the local switch cross-points. However, the inter-layer cross-point is modified to enable the class-based scheme.

Fig. 7 shows a *single* cross-point within an inter-layer switch for an input binned 4-channel 4-layer 64-radix configuration. This configuration has 13 cross-points (corresponding to 12 L2LCs and a local input) in a sub-block. Each cross-point provides connectivity for an L2LC, which in turn is associated with four primary inputs. For each of these four primary inputs, a thermometer *class counter* is placed within the inter-layer switch cross-point. We find empirically that three classes provide reasonable fairness for a 64-radix *Hi-Rise* switch, hence we use a thermometer counter with the following sequence {00,01,11}.

The counter for a primary input, keeps track of how many times that input won the arbitration for the final output. During the arbitration phase the counter value of the primary input which wins the L2LC, is chosen using a multiplexer ($Mux1$). This counter value is used for setting up other multiplexer as shown in Fig. 7.

The arbitration circuit shown in the figure enables class based arbitration along with LRG tie breaking in a single cycle. The output lines are reused as priority lines during the arbitration phase. The priority lines are grouped class-wise, where each group has priority lines for *each* of the 13 L2LCs. Priority Class '00' uses wires 0-12, priority class '01' uses wires 13-25, and priority class '11' uses wires 26-38 as shown in the figure.

Each cross-point has three Priority Select Multiplexers (PSMs) as shown in Fig. 7. The PSMs apply '1' to all priority lines belonging to a lower priority class, so that any request from a lower-priority class is inhibited. The PSMs apply '0' to all priority lines belonging to a higher priority class, so that it does not affect the arbitration for the higher priority class. The PSM applies the LRG priority vector to the priority lines belonging to its own class.

The arbitration circuit thus allows L2LC with a higher priority class to pull-down the priority lines being polled by the L2LCs with a lower priority class. The L2LC in the highest priority class thus wins. However, if multiple L2LCs in the highest priority class are requesting, then they pull-down priority lines of L2LCs with lower LRG priority in their own class.

Each L2LC polls one priority line in each of the three priority class. The multiplexer ($Mux2$) as shown in Fig. 7 is used to select one of the three lines based on class counter. The polled value goes to a sense-amplifier enabled latch, which is the connectivity bit. Once this bit is set, the cross-point connects the input data to the output lines. The winning primary input also increments its corresponding counter.

C. Clocking of Hi-Rise Switch

As shown in Fig. 8, the *Hi-Rise* switch uses two-phase clocking. In the first phase the local switch evaluates and transmits the outputs to the inter-layer's inputs.

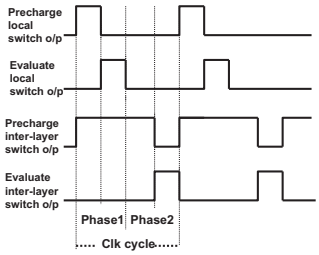

Figure 8. 2-Phase clocking of *Hi-Rise* Switch. Phase 1: Evaluates local switch; Phase 2: Evaluates inter-layer switch

The inter-layer switch stays pre-charged until the outputs of all the local switches have been evaluated and stabilized. In the second phase, the inter-layer switch evaluates and generates the final output. Intermediate outputs are not latched at the local switch outputs.

D. Physical Implementation

The cross-point layout for the circuit schematic shown in Fig. 6, has horizontal input bus metal wires, and vertical output bus metal wires, with logic underneath. The 3D local switch cross-points and inter-layer switch cross-points for baseline have fewer priority bits than in a 2D cross-point, so the logic area is significantly lower. The area is thus wire limited and the switch can be extended to higher radices before the logic becomes dominant. For CLRG arbitration, due to the additional counters in each cross-point, the inter-layer cross-point's gate count is comparable to the 2D cross-point. To reduce switch area, wires are stacked using two metal layers in each direction. To reduce coupling between wires, double pitch spacing is used.

The TSV used for evaluation has a 0.8μ minimum pitch with 0.2fF feed-through capacitance and $1.5ohm$ resistance [16]. Section VI-C studies TSV parameters and the impact of TSV size on the performance and area of the proposed *Hi-Rise* switch. The TSVs are all located in the local switch. The 4-channel input binned configuration has only four cross-points out of a column of 16, leaving plenty of empty space to place the TSVs without any area increase.

V. Methodology

To evaluate the various interconnect performance characteristics, a cycle accurate network simulator is used. To accurately model the circuit implementation, C models are written for each of the different switch configurations and arbitration schemes described in Sections III-A and III-B. The baseline design is a 2D 64×64 switch.

The synthetic traffic patterns used to evaluate the various switch configurations are uniform random, hot spot and bursty. Custom synthetic traffic patterns are also used to evaluate specific corner cases and adversarial cases for the proposed switch configurations. For synthetic patterns, the simulator uses 4 virtual channels at each port with a buffer depth of 4 flits per virtual channel. Each flit has a size of 128 bits to match the databus width, and 4 flit packets have been used for simulations.

Spice models for both the baseline arbitration scheme and the CLRG arbitration scheme are created based on their cross-point implementation. The spice models for the switch are verified against the 2D *Swizzle-Switch* silicon results. The spice models are in a commercial $32\ nm$ SOI technology. These models are then used to determine the area, speed and energy for the entire switch. The spice netlist accurately models the effect of wire routing, with appropriate length wire models of the correct metal layers used. The spacing between the wires is double-pitched to avoid capacitive coupling. Physical implementation details, like using multiple metal layers stacking for input and output routing to reduce wire lengths are also considered. The spice model accounts for the capacitive loading of the TSVs, and also the routing to and from the TSV. The TSV parameters and the Spice PVT conditions used for evaluations are as shown in the Table II.

Table II. Spice condition and TSV Parameters

Spice Conditions	Process = Typical	Temperature = 27C	Voltage = 1V
TSV Parameters	Pitch = $0.8\ \mu m$	Feed-Through Cap. = $0.2\ fF$	Resistance = $1.5\ ohm$

To run real application workloads, a trace-driven, cycle-accurate many-core simulator [17] is integrated with a system built out of a single *Hi-Rise* switch, cores, caches and memory controller models. The system parameters used for application workloads is shown in Table III. A front-end functional simulator based on Pin [18] is used to collect instruction traces from applications, which are then fed into the cycle-level simulator. We study a diverse set of benchmarks, including SPEC CPU2006 [19] benchmarks, and four commercial workload traces (`sap`, `tpcw`,`sjbb`, `sjas`).

Table III. Processor configuration for application workloads

Cores	64 cores , 2-way out-of-order, 2 GHz frequency
L1 Caches	32 KB per-core, private, 4-way set associative, 64B blocks, 2-cycle latency, split I/D caches, 32 MSHRs
L2 Caches	64 banks, 256KB per bank, shared, 16-way set associative, 64B block size, 6-cycle latency, 32 MSHRs
Main Memory	8 on-chip memory controllers, 4 DDR channels each @16GB/s, up to 16 outstanding requests per core, 80ns access latency

VI. Results

The proposed 3D switch has several design parameters that can be tuned to gain better performance, increased scalability, and reduced implementation cost. In Section VI-A, we first study the datapath of the proposed 3D switch and find the optimal point of operation with respect to both the network characteristics and the implementation cost. In Section VI-B, we evaluate the different built-in arbitration schemes. Section VI-C analyzes the sensitivity to the TSV technology. Finally, we present the results for the application workloads in Section VI-D.

A. Analysis of 3D Switch Datapath Parameters

This section will first discuss how we optimize *Hi-Rise* switch for speed. We then compare network characteristics for uniform random(UR) traffic. The goal is to find a configuration with high saturation throughput, low latency and high speed of operation.

In the proposed 3D switch, frequency is a function of the radix of the switch and the number of layers stacked. The L2LC multiplicity 'c' is also a factor, as increased 'c' causes both the local switches and inter-layer switches to grow in size. Fig. 9(a) shows the frequency for different radices of

a 4-layered 3D switch, and the 2D *Swizzle-Switch*. The 2D switch has a better frequency at low radix, as the overheads incurred by the hierarchical architecture makes the 3D switch slow. Beyond radix 32, all 3D configurations have a better speed than 2D. As the radix increases the frequency gap widens, making the 3D switch more favorable. As radix increases, the channel multiplicity also becomes less of a factor, as can be seen from the converging 1, 2 and 4 channel frequency plots.

The number of silicon layers stacked is another factor that changes the frequency significantly. At a low number of stacked layers, the switches on each layer are still large, so the frequency will be low. However, if we have too many layers, the numbers of L2LCs increase, and become the dominating factor. Therefore, the number of stacked layers in a switch has an optimal point. As seen in Fig. 9(b), for a 64-radix 3D switch the frequency is maximum in the range of 3 to 5 layers and then decreases on either side. At small radices, the optimum number of layers required is lower, whereas for higher radices the optimum point shifts towards higher number of stacked layers. We use a 3D 64-radix switch as a fair comparison, because the 2D *Swizzle-Switch* scales well until 64-radix. As seen in Fig. 9(b), for 64-radix the optimal number of stacked layers is 4. The 4-layer 64-radix 3D switch can still have different channel multiplicity numbers. Lower channel multiplicity will have lower overhead, but may not provide sufficient throughput.

The network throughput for channel multiplicity of 1, 2 and 4 are listed in Table IV. The latency curves are shown in Fig. 10. The 3D one-channel configuration performs poorly and saturates at very low injection rates. The configuration with channel multiplicity of 2 is only 19% worse than a 2D flat switch's throughput, while the configuration with channel multiplicity of 4 has 18% better throughput than the 2D switch. The proposed 4-channel 4-layer 64-radix 3D switch has a saturation throughput of 21.42 packets/ns or 10.97 Tbps for uniform random traffic. Also, the zero-load latency for proposed 3D configurations is about 20% better than 2D. So even at low injection rates the 3D will outperform the 2D switch. The naive folded implementation, on the other hand, has 7% less saturation throughput than a 2D flat switch.

The energy consumed per 128-bit transaction is also an important metric for switch performance. The compactness of the *Hi-Rise* switch makes it more energy efficient. Fig. 9(c) shows the energy per transaction as the radix increases. The 3D switch energy increases at a more gradual slope as compared to a 2D switch, allowing it to have a significantly higher radix switch for iso-energy.

The implementation cost for the various channel configurations is shown in Table IV. The proposed 3D switch's hierarchical structure leads to much smaller switches, and hence the large implementation cost benefits over a 2D *Swizzle-Switch*. These switches are smaller not just in their dimension but also in the gate count. The 4-Channel 3D switch has a 40% lower energy requirement than the 2D switch, and occupies 33% less area. Also, the implementation cost of the 4-Channel 3D switch is not significantly higher than the 2-Channel 3D switch. From both implementation cost and channel multiplicity traffic study, we choose the 4-channel 4-layer 64-radix *Hi-Rise* switch as the optimal configuration for all further analysis.

B. Analysis of 3D Switch Arbitration Schemes

In Section III-B three arbitration schemes were discussed: baseline layer-to-layer LRG (L2L LRG); Weighted LRG (WLRG); and, Class-based LRG (CLRG). The goal of the proposed arbitration schemes is to make the switch fair. In this section we present the results of analyzing fairness for the various traffic patterns. We also present the implementation cost for these arbitration schemes in hardware.

Hotspot traffic helps bring out the fairness issue in the baseline L2L LRG. Hotspot traffic involves all inputs requesting the same output. The pattern used in this experiment involves all inputs from layers 1, 2, 3 and 4, requesting for output 63. Fig. 11(a) shows the average latency for inputs 0 to 63 in cycles. The load rate used in this experiment is 80% of the saturation load rate for hotspot traffic.

In 3D 4-channel 4-layer configuration, any inter-layer sub-block has one connection for local intermediate output and twelve L2LC connections. This causes the 3D L2L LRG arbitration to show a wide deviation between the latency for local inputs, and the latency for other layer inputs. Since, all requests are for the same output in hotspot traffic, the local intermediate output is arbitrated for by 16 primary inputs, while each L2LC is arbitrated for by 4 primary inputs only. Thus, L2L LRG effectively allots only 1/4th of the bandwidth to the local intermediate output as compared to other layers. This is evident from the high latency for the local inputs {48 to 63} in the Fig. 11 (a).

In the CLRG scheme the unfairness is resolved. Initially all inputs have a count of 0, i.e., the highest priority class. The other layer inputs get through faster initially, as only 4 primary inputs contend for a L2LC. But as they keep winning, they are relegated to a lower priority class, thus elevating the priority of the non-serviced requests from the local layer. In the case of hotspot traffic, every primary input will reach a count of 1 before anyone gets to transmit again. Hence, class-based arbitration behaves similar to flat LRG.

Comparison of the average throughput for different arbitration schemes with uniform random traffic is shown in Fig. 11 (b). For uniform random traffic, even the 3D L2L LRG arbitration scheme behaves in an unbiased manner. Hence, the performance for uniform random traffic mainly depends on the frequency of operation for the switches. The 3D L2L LRG, due to its design simplicity, is able to run marginally faster than the CLRG arbitration. Also, all the 3D arbitration schemes are still considerably faster than a 2D

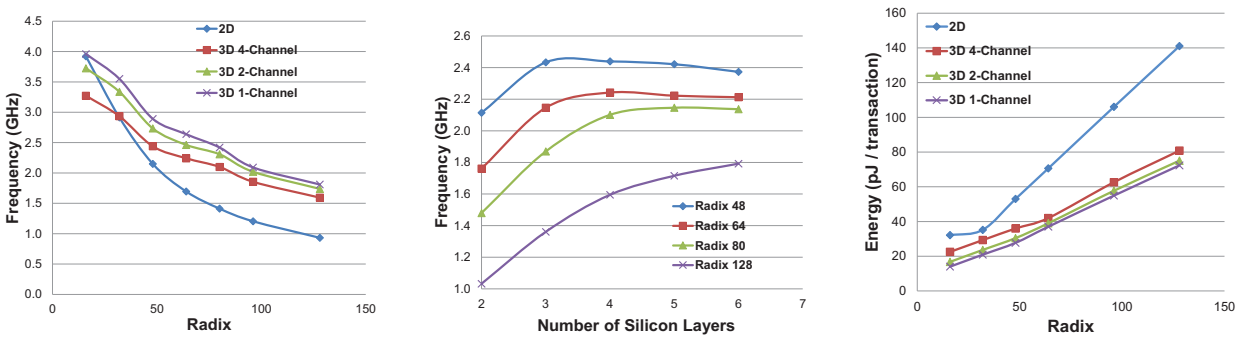

Figure 9. (a) Frequency vs radix, (b) Frequency vs number of silicon layers stacked and (c) Energy per transaction (128-bit) for 2D and 3D Switch.

Table IV. IMPLEMENTATION COST OF DIFFERENT SWITCH IMPLEMENTATIONS FOR 64-RADIX. THE 3D SWITCHES HAVE 4-LAYERS. THE CONFIGURATION COLUMN HAS LOCAL SWITCH SIZE, FOLLOWED BY INTER-LAYER SWITCH. EACH TRANSACTION IS 128-BITS.

Design	Configuration	Area (mm^2)	Frequency (GHz)	Energy/transaction ($pJ/trans$)	Throughput ($Tbps$)	#TSVs
2D	64×64	0.672	1.69	71	9.24	0
3D Folded	[16×64]×4	0.705	1.58	73	8.86	8192
3D 4-Channel	[(16×28), 16·(13×1)]×4	0.451	2.24	42	10.97	6144
3D 2-Channel	[(16×22), 16·(7×1)]×4	0.315	2.46	39	7.65	3072
3D 1-Channel	[(16×19), 16·(4×1)]×4	0.247	2.64	37	4.27	1536

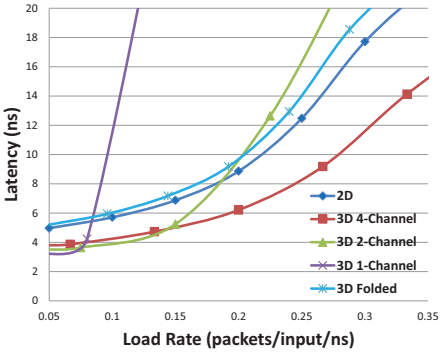

Figure 10. Latency of 2D and 3D multi-channel configurations for UR traffic.

flat switch, as seen from the frequency numbers in Table V. Thus, the throughput of CLRG is slightly lower than the 3D L2L LRG, but compared to a 2D switch, they have 15% better throughput. The latency of all the three schemes is very similar, and all the schemes have a zero-load latency that is about 20% better than a 2D *Swizzle-Switch*.

We study the latency and throughput for the adversarial traffic pattern, which was used as the example in Section III-B. The pattern consists of five requesting inputs, four inputs {3, 7, 11, 15} from L1, and input {20} from L2. All five inputs are requesting output 63 on L4. For this pattern, the L2L LRG shows a wide disparity between throughputs of input 20 versus the throughput of the other four inputs. This is shown in Fig. 11 (c). Both the WLRG and the CLRG arbitration schemes are able to resolve this bias as explained in Section III-B.

A *pathological case* for the 3D switch is when we have only *inter-layer* traffic, but no *within-layer* traffic. In this case, the throughput is limited by the bandwidth available through the L2LCs between any two layers. The throughput for such traffic is not improved by the different arbitration schemes. The worst case scenario is, all the four inputs using the same L2LC, request for different outputs on another layer. In this corner case, the throughput of the 3D switch can get limited up to 1/4th of the flat 2D switch.

C. TSV Technology Parameters

In Section VI-A effects of design parameters like the number of stacked layers, and number of L2L channels were discussed. Another important consideration that can affect both the implementation cost and the performance is the TSV technology being used. The TSV technology used in this switch is a high-end $0.8um$ pitch TSV. The area and switch delay for a less advanced TSV technology will be more, because of the bigger pitch and higher wire parasitics. However, TSV pitch has constantly been going down as 3D integration evolves. The advancement of technology will lead the 3D switch to become effective even at low radices.

Fig. 12 shows the area increase with TSV pitch. This area increase is attributed to the fact that a TSV punches through the silicon layer, rendering that area useless. The area increase also factors in the routing to and from the TSVs. The increase in area and capacitive loading for large pitched TSVs in less advanced technologies causes the delay to increase. Even with an additional 25% pitch, Hi-Rise area increases by only 1.67%, and frequency falls by 1.8%. Additionally, Tezzaron [16] uses Tungsten TSVs instead of

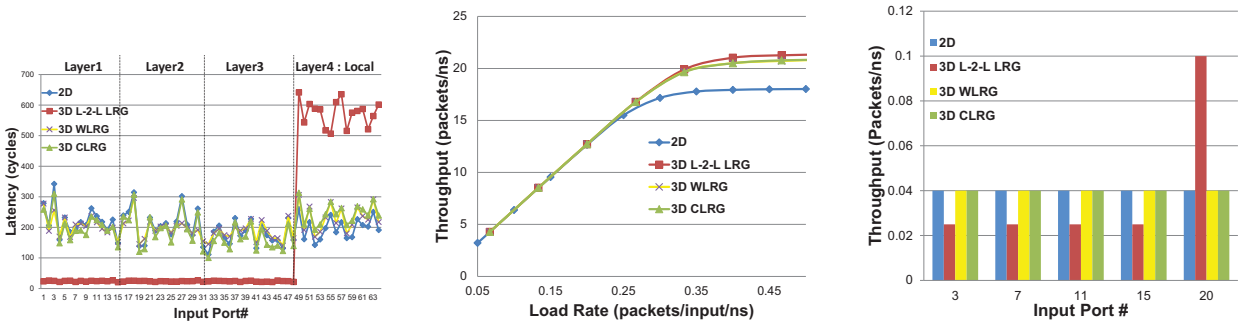

Figure 11. (a) Latency of each input for hotspot traffic. All inputs 0 to 63, requesting o/p 63 on Layer 4. (b) Throughput of arbitration schemes for UR traffic. (c) Throughput of requesting inputs for baseline's adversarial traffic.

Table V. IMPLEMENTATION COST OF DIFFERENT SWITCH ARBITRATION VARIANTS FOR 64-RADIX. THE 3D SWITCHES HAVE 4-CHANNEL 4-LAYERS. WLRG NOT SHOWN AS ITS IMPLEMENTATION IS INFEASIBLE.

Design	Configuration	Area (mm^2)	Frequency (GHz)	Energy/transaction ($pJ/trans$)	Throughput ($Tbps$)	#TSVs
2D	64X64	0.672	1.69	71	9.24	0
3D L-2-L LRG	[(16×28), 16·(13×1)]×4	0.451	2.24	42	10.97	6144
3D CLRG	[(16×28), 16·(13×1)]×4	0.451	2.2	44	10.65	6144

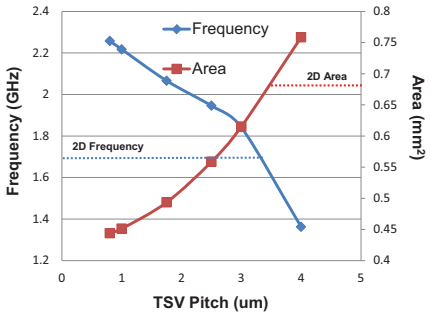

Figure 12. Sensitivity of frequency and area to TSV pitch for *Hi-Rise* 64-radix 4-Channel 4-layered configuration

copper, which has a matching expansion coefficient with silicon, hence keep-out-zone is negligible. Another feature of Hi-Rise topology is that it can use clustered TSVs for the Layer-to-Layer Channels, amortizing the affect of keep-out-zones, for other TSV technologies.

D. Application Results

In this section, we evaluate the proposed *Hi-Rise* switch for real application traffic. For this, a 64-core system using a single switch as the interconnect fabric is created as discussed under the methodology section. The two systems used for comparison are identical, except that one has a 2D flat switch as the interconnect, while the other uses the *Hi-Rise* 4-channel 4-layer switch with CLRG built into it.

To evaluate the effect on performance, eight different multi-programmed workloads are simulated. Each workload consists of six applications, with multiple application instances used to construct the workload as shown in the Table VI. The applications' allocation is done randomly, and is *oblivious of the layer-to-layer dependencies* in the switch.

The last column of Table VI shows the normalized system speedup of the proposed 3D switch over a 2D *Swizzle-Switch*. The 3D switch outperforms the 2D switch by 8% on an average. The 3D switch provides better speedup for workloads with higher cache miss rates. For Mix8, which has the largest MPKI amongst all workloads, the proposed 3D switch shows a 15% performance advantage.

E. Discussion

2D *Swizzle-Switch* [12] has been compared to other topologies like mesh and *Swizzle-Switch* enhanced flattened butterfly. We chose the 2D *Swizzle-Switch* for comparison, as its power is 33% better than mesh and 28% better than flattened butterfly. *Hi-Rise* further improves over the 2D *Swizzle-Switch* power by about 38%, giving us about 58% power savings over flattened butterfly. The system speedup of *Hi-Rise* over flattened butterfly is approximately 13%.

In this section, we also briefly discuss composing Hi-Rise switches to form larger topologies with 1000 cores (kilo-core). Future kilo-core systems cannot use existing low-radix networks due to scalability issues. To get around this, prior designs have proposed high-radix topologies with concentration [4], [5]. This helps reduce the number of routers in the network in addition to reducing the average hop count. *Hi-Rise*, or other true 3D switch designs, can also be used to make NoC topologies for 3D chips like the one shown in Fig. 13. The topology is a *2D mesh of 3D switches*. This allows routing algorithms to be XY dimensionally ordered, while the 3D switch can provide the adaptable Z dimension routing, leading to optimal utilization of the L2LC. Layer-aware routing algorithms that minimize the traversal of traffic in the vertical direction will also help alleviate the L2LC bottleneck problems within the switch.

Table VI. BENCHMARKS USED TO CONSTRUCT EIGHT MULTI-PROGRAMMED WORKLOADS FOR A 64-CORE PROCESSOR SYSTEM USING A SINGLE 64-RADIX SWITCH. NUMBERS IN PARENTHESIS INDICATE THE NUMBER OF APPLICATION INSTANCES USED TO CONSTRUCT THE WORKLOAD. THE AVERAGE MISSES-PER-KILO-INSTRUCTION (MPKI) PER CORE IS THE SUM OF THE BENCHMARK'S L1-MPKI AND L2-MPKI, WHICH CORRESPONDS TO THE NETWORK LOAD FOR THE WORKLOADS.

Mix							avg. MPKI	Speedup
Mix1	milc (11)	applu (11)	astar (10)	sjeng (11)	tonto (11)	hmmer (10)	15.0	1.02
Mix2	sjas (11)	gcc (11)	sjbb (11)	gromacs (11)	sjeng (10)	xalan (10)	21.3	1.04
Mix3	milc (11)	libquantum (10)	astar (11)	barnes (11)	tpcw (11)	povray (10)	33.3	1.06
Mix4	astar (11)	swim (11)	leslie (10)	omnet (10)	sjas (11)	art (11)	38.4	1.06
Mix5	mcf (11)	ocean (10)	gromacs (10)	lbm (11)	deal (11)	sap (11)	52.2	1.08
Mix6	mcf (10)	namd (11)	hmmer (11)	tpcw (11)	omnet (10)	swim (11)	58.4	1.09
Mix7	Gems (10)	sjbb (11)	sjas (11)	mcf (10)	xalan (11)	sap (10)	66.9	1.16
Mix8	milc (11)	tpcw (10)	Gems (11)	mcf (11)	sjas (11)	soplex (10)	76	1.15

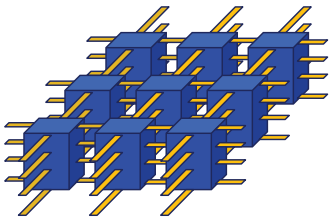

Figure 13. A 2D mesh NoC topology composed of 3D *Hi-Rise* switches for 3D chips.

VII. RELATED WORK

Prior works have shown that high-radix switches can be used to construct low latency networks [4], [5], [20], [21], [22], [23]. Kim et al. [22] proposed several optimizations to improve the scalability of switches. The optimizations include breaking down the arbitration into multiple local and global stages, and hierarchical crossbars with intermediate buffering. Our proposed *Hi-Rise* is a *3D high-radix switch* composed of two switches that does not have intermediate buffering and has deterministic datapath connections, allowing it to either arbitrate or transmit data in a single-cycle.

As 3D systems become more main stream, 3D integration becomes extremely important in interconnect fabrics. However, most of the 3D switches proposed in recent years have been low radix switches [7], [8], [9], [10]. Xu et al. [10] explored $4 \times 4 \times 4$ and $4 \times 4 \times 5$ 3D switch to reduce implementation cost. Kim et al. [9] proposed a low-radix 3D switch which was customized for dimension ordered routing in mesh topologies.

Lewis et al. [24] proposes a folded 3D crossbar and 3D Multistage Interconnect Networks (MINs). It optimizes the folded 3D crossbar by adding switches at each output of different stack layers to cut down the critical path. Unlike [24], in *Hi-Rise* datapath, only the local layer has a unique bus to each sub-block, while all inter-layer routing is shared among the sub-blocks. This considerably eases the routing and TSV requirements, especially for a wide data-bus. [24] partitions MIN networks effectively to reduce both wiring density and wiring length in 3D. But, MIN networks made up of 2x2 switches, require many stages for high-radix, and have contentions. *Hi-Rise* datapath has only two heterogeneous stages, each of which is composed of an efficient, contention-free switch.

Several arbitration policies are possible for a switch arbiter [14]. Policies like Longest Queue First (LQF) [25] and Oldest Cell First (OCF) [25] have been used. WLRG is similar to LQF, which is also based on higher priority for higher number of requestors. However, as discussed, the implementation cost of WLRG makes it infeasible. Similarly, [26] uses distance based weights. OCF chooses oldest request based on timestamps, which requires a prohibitively expensive comparison, especially for on-chip high-radix switch with single cycle arbitration. [27] also uses an age-based arbitration. For high-radix switches, Ping-Pong Arbiters (PPA) [28] have been used which combine small arbiters in a comparison tree. These are also difficult to integrate within the datapath. The CLRG arbitration fits into the datapath itself, reuses the output lines, utilizes embedded self-updating priorities and inhibit logic for arbitration. This reduces the implementation cost significantly, and allows scalability to high-radices with a single-cycle arbitration.

Allocators [13], [29], [30], [31] utilize multi-stage arbiters to maximize the output bandwidth utilization by matching the requests from the virtual channel buffers to the switch outputs. Allocation schemes like hierarchical switches, also try to compose multi-level arbitration schemes. Allocation policies utilize different combinations of round-robin arbitration [13], [30], [31]. For example, an iteration of iSLIP [31] updates the round-robin priority at the pre-final stage in a multi-stage arbitration, only if the input wins at the final stage. A single iteration of iSLIP is similar to the baseline L-2-L LRG we discussed before and does not solve the fairness issues. In Backlog Weighted Round-Robin (BWRR), proposed for hierarchical switches [32], a backlog signal is passed from the first stage to the second stage. The second stage does not update its priority if the backlog signal for the winner is high. This technique incurs similar overheads as WLRG. The CLRG arbitration is also a multi-stage arbitration scheme, which trades-off implementation complexity and fairness. Our proposed *class* based division of primary inputs, allows maintaining a coarse-grained LRG at the inter-layer switch for each primary input.

VIII. Conclusion

The processor industry is moving towards 3D integration and more cores per chip. This is creating the need for interconnects with features to exploit the potential of 3D integration.

This paper presents *Hi-Rise*, a fast, high-bandwidth, area-energy efficient, high-radix, 3D switch, with single-cycle built-in arbitration. The proposed *Hi-Rise* switch adopts a hierarchical architecture with two internal switches per layer and dedicated layer-to-layer channels. The inter-layer switch on each layer makes the proposed solution a true 3D switch which connects inputs and outputs across different silicon layers. The paper proposes an integrated arbitration scheme to resolve unfairness in a hierarchical 3D switch. The proposed Class-based Least Recently Granted (CLRG) scheme is able to provide fairness comparable to that of a flat 2D switch with Least Recently Granted arbitration.

This is the first paper which presents an efficient 3D high-radix switch design. The proposed 3D switch is evaluated for different radices, number of stacked layers and different TSV technology parameters. A 64-radix, 128-bit width, 4-layer *Hi-Rise* evaluated on a 32nm technology has a throughput of 10.65 Tbps for uniform random traffic, which marks a 15% improvement in throughput over a 2D design, along with a 33% area reduction, 20% latency reduction, and 38% reduction in energy per transaction.

Acknowledgment

This work was partially sponsored by ARM, NSF award CCF-1256203 and DARPA agreement HR0011-13-2-000.

References

[1] J. Howard, S. Dighe *et al.*, "A 48-core ia-32 message-passing processor with dvfs in 45nm cmos," in *ISSCC*, 2010.

[2] D. Wentzlaff, P. Griffin *et al.*, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, no. 5, pp. 15–31, 2007.

[3] S. Borkar, "Thousand core chips: a technology perspective," in *DAC-44*, 2007.

[4] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *MICRO-40*, 2007.

[5] N. Abeyratne, R. Das *et al.*, "Scaling towards kilo-core processors with asymmetric high-radix topologies," in *HPCA-19*, 2013.

[6] D. Fick, R. G. Dreslinski *et al.*, "Centip3de: A 3930dmips/w configurable near-threshold 3d stacked system with 64 arm cortex-m3 cores," in *ISSCC*, 2012.

[7] D. Park, S. Eachempati *et al.*, "MIRA : A Multilayered Interconnect Router Architecture," in *ISCA-35*, 2008.

[8] F. Li, C. Nicopoulos *et al.*, "Design and management of 3d chip multiprocessors using network-in-memory," in *ISCA*, 2006.

[9] J. Kim, C. Nicopoulos *et al.*, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," *ISCA-34*, 2007.

[10] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, "A low-radix and low-diameter 3d interconnection network design," in *HPCA*, 2009.

[11] S. Satpathy, K. Sewell *et al.*, "A 4.5tb/s 3.4tb/s/w 64x64 switch fabric with self-updating least recently granted priority and quality of service arbitration in 45nm cmos," in *ISSCC*, 2012.

[12] K. Sewell, R. Dreslinski *et al.*, "Swizzle-switch networks for many-core systems," in *JETCAS*, 2012.

[13] N. W. McKeown, *Scheduling algorithms for input-queued cell switches, PhD Thesis*, 1992.

[14] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[15] S. Satpathy, R. Das, R. Dreslinski, D. Sylvester, T. Mudge, and D. Blaauw, "High radix self-arbitrating switch fabric with multiple arbitration schemes and quality of service," in *DAC-49*, 2012.

[16] "3D Integration: New Opportunities for Speed, Power and Performance," www.tezzaron.com/about/papers/Tezzaron-Presentation-CASS-020712-dist-2.pdf.

[17] R. Das, O. Mutlu, T. Moscibroda, and C. R. Das, "Application-aware prioritization mechanisms for on-chip networks," in *MICRO-42*, 2009.

[18] H. Patil, R. Cohn *et al.*, "Pinpointing representative portions of large intel itanium programs with dynamic instrumentation," in *MICRO-37*, 2004.

[19] J. L. Henning, "Spec cpu2006 benchmark descriptions," in *ACM SIGARCH Computer Architecture News 34.4*, 2006.

[20] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *ISCA-34*, 2007.

[21] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *35th International Symposium on Computer Architecture (ISCA)*, 2008.

[22] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta, "Microarchitecture of a high-radix router," in *ISCA-32*, 2005.

[23] S. Scott, D. Abts, J. Kim, and W. J. Dally, "The blackwidow high-radix clos network," in *ISCA-33*, 2006.

[24] D. L. Lewis, S. Yalamanchili, and H.-H. Lee, "High performance non-blocking switch design in 3d die-stacking technology," in *ISVLSI*, 2009.

[25] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in *Communications, IEEE Transactions on*, 1999.

[26] M. M. Lee, J. Kim, D. Abts, M. Marty, and J. W. Lee, "Probabilistic distance-based arbitration: Providing equality of service for many-core cmps," in *MICRO-43*, 2010.

[27] D. Abts and D. Weisser, "Age-based packet arbitration in large-radix k-ary n-cubes," in *SC*, 2007.

[28] H. J. Chao, C. Lam, and X. Guo, "A fast arbitration scheme for terabit packet switches," in *GLOBECOM*, 1999.

[29] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," in *ACM Transactions on Computer Systems (TOCS)*, 1993.

[30] D. N. Serpanos and P. Antoniadis, "Firm: A class of distributed scheduling algorithms for high-speed atm switches with multiple input queues," in *INFOCOM*, 2000.

[31] N. McKeown, "The islip scheduling algorithm for input-queued switches," *Networking, IEEE/ACM Trans. on*, 1999.

[32] J. A. Jum, S. H. Byun, B. J. Ahn, S. Y. Nam, and D. K. Sung, "A two-dimensional scalable crossbar matrix switch architecture," in *IEEE International Conf. on Communications*, 2003.